

Chapter 7.8

A Model-Based Approach for Diagnosing Fault in Web Service Processes

Yuhong Yan

Concordia University, Canada

Philippe Dague

University Paris-Sud 11, France

Yannick Pencolé

LAAS-CNRS, France

Marie-Odile Cordier

IRISA, France

ABSTRACT

Web service orchestration languages are defined to describe business processes composed of Web services. A business process can fail for many reasons, such as faulty Web services or mismatching messages. It is important to find out which Web services are responsible for a failed business process because we could penalize these Web services and exclude them from the business process in the future. In this paper, we propose a model-based approach to diagnose the faults in a Web service-composed business process. We convert a Web service orchestration language,

BPEL4WS, into synchronized automata, so that we have a formal description of the topology and variable dependency of the business process. After an exception is thrown, the diagnoser can calculate the business process execution trajectory based on the formal model and the observed evolution of the business process. The faulty Web services are deduced from the variable dependency on the execution trajectory.

INTRODUCTION

Web services not only function as middleware for application invocation and integration, but

also function as a modeling and management tool for business processes. In a Service Oriented Architecture paradigm, a business process can be composed of Web services distributed over the Internet. This kind of business processes can be flexible and optimal by using the best services from multiple companies. Various Web service process description languages are designed by standard bodies and companies. Among them, Business Process Execution Language for Web Service (BPEL4WS, denoted as BPEL after) (Andrews, Curbera, Dholakia, Goland, et al., 2003) is the de facto standard used to describe an executable Web service process. In this article, we study the behavior of a business process described in BPEL.

As any other systems, a business process can fail. For a Web service process, the symptom of a failure is that exceptions are thrown and the process halts. As the process is composed of multiple Web services, it is important to find out which Web services are responsible for the failure. If we could diagnose the faulty Web services, we could penalize these Web services and exclude them from the business process in the future. The current throw-and-catch mechanism is very preliminary for diagnosing faults. It relies on the developer associating the faults with exceptions at design time. When an exception is thrown, we say certain faults occur. But this mechanism does not guarantee the soundness and the completeness of diagnosis.

In this article, we propose a model-based approach to diagnose faults in Web service processes. We convert the basic BPEL activities and constructs into synchronized automata whose *states* are presented by the values of the *variables*. The process changes from one state to another by executing an *action*, for example, assigning variables, receiving or emitting messages in BPEL. The emitting messages can be a triggering *event* for another service to take an action. The diagnosing mechanism is triggered when exceptions are thrown. Using the formal model and the runtime

observations from the execution of the process, we can reconstruct the unobservable trajectories of the Web service process. Then the faulty Web services are deduced based on the variable dependency on the trajectories. Studying the fault diagnosis in Web service processes serves the ultimate goal of building self-manageable and self-healing business processes.

This article is organized as follows: the Advanced Fault Management for Web Service Processes section analyzes the fault management tasks in Web service processes and motivates the use of Model-based Diagnosis (MBD) for Web services monitoring and diagnosis; The Principle of Model-based Diagnosis for Discrete Event Systems section presents the principles for MBD; the Modeling Web Service Processes with Discrete-Event Systems section formally defines the way to generate an automaton model from a BPEL description; the Model-based Diagnosis for Web Service Processes section extends the existing MBD techniques for Web service monitoring and diagnosis; the Related Work and Discussion section is the related work, and lastly is the Conclusion section.

ADVANCED FAULT MANAGEMENT FOR WEB SERVICE PROCESSES

A Web service process can run down for many reasons. For example, a composed Web service may be faulty, an incoming message mismatches the interface, or the Internet is down. The *symptom*¹ of a failed Web service process is that *exceptions* are thrown and the process is halted. The current fault handling mechanism is throw-and-catch, similar to programming languages. The *exceptions* are thrown at the places where the process cannot be executed. The *catch* clauses process the exceptions, normally to recover the failure effects by executing predefined actions.

The throw-and-catch mechanism is very preliminary for fault diagnosis. The exception

22 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/model-based-approach-diagnosing-fault/37725

Related Content

New Entropy Based Distance for Training Set Selection in Debt Portfolio Valuation

Tomasz Kajdanowicz, Slawomir Plamowski and Przemyslaw Kazienko (2012). *International Journal of Information Technology and Web Engineering* (pp. 60-69).

www.irma-international.org/article/new-entropy-based-distance-training/70386

Time Effective Cloud Resource Scheduling Method for Data-Intensive Smart Systems

Jiguang Duan, Yan Li, Liying Duan and Amit Sharma (2022). *International Journal of Information Technology and Web Engineering* (pp. 1-15).

www.irma-international.org/article/time-effective-cloud-resource-scheduling-method-for-data-intensive-smart-systems/306915

Algorithm for Secure Hybrid Cloud Design Against DDoS Attacks

Akashdeep Bhardwaj and Sam Goundar (2018). *International Journal of Information Technology and Web Engineering* (pp. 61-77).

www.irma-international.org/article/algorithm-for-secure-hybrid-cloud-design-against-ddos-attacks/209721

Behavioral Intentions to Adopt Technological Innovations: The Role of Trust, Innovation and Performance

Vanessa Ratten (2016). *Web-Based Services: Concepts, Methodologies, Tools, and Applications* (pp. 1496-1508).

www.irma-international.org/chapter/behavioral-intentions-to-adopt-technological-innovations/140862

What is Different about Web and Software Projects?

Emilia Mendes (2008). *Cost Estimation Techniques for Web Projects* (pp. 1-24).

www.irma-international.org/chapter/different-web-software-projects/7160